

重庆邮电大学

# 移动互联网应用开发 1 作品设计报告

学年学期： 2022-2023 学年 1 学期

课程名称： 移动互联网应用开发1

学生学院： 通信与信息工程学院

专业班级： 01042002

学生姓名： 黄廷峰

学生学号： 2020210497

作品名称： mini 音乐视频播放器

重庆邮电大学教务处制

## 目录

一、作品实现目标 .....	3
二、作品背景.....	3
三、作品设计内容和方法.....	4
1.主要设计内容.....	4
2.解决的关键问题.....	4
1) 注册账号问题 .....	4
2) 页面布局问题 .....	5
3) Input 输入检测问题.....	6
4) 页面跳转问题 .....	8
3.主要设计方法及采用的辅助工具.....	9
四、系统功能模块描述及概要设计 .....	9
1) 登陆模块 .....	9
2) 注册模块 .....	10
3) 修改密码模块 .....	10
4) 我的音乐视频模块.....	10
5) 视频播放器模块 .....	11
6) 音乐播放器模块 .....	11
五、系统详细实现与测试结果分析 .....	12
1) 登陆模块 .....	12
2) 注册模块 .....	15
3) 修改密码模块 .....	19
4) 视频播放器模块 .....	22
5) 音乐播放器模块 .....	24
6) 我的音乐视频模块.....	26
六、问题与总结 .....	27

## 一、 作品实现目标

Mini 音乐视频播放器通过存储 API 以及表单 API 并调用本地存储可以实现用户登录验证，用户的账号注册，以及修改密码等操作，来实现对用户的登陆验证，最终进入到音乐视频的播放页面。通过媒体 API 以及画布 API 可以实现音乐名称的滚动，以及音乐视频的上下首播放，暂停，调节音量，全屏，画中画等功能，还可以查看自己的添加的音乐视频。Mini 音乐视频播放器主要通过 html，css，js 实现了一些简单的登录注册修改密码，视频音乐播放，查看我的音乐视频等功能，不仅可以实现对用户的验证，也可以给用户带来不错的音乐视频体验。

## 二、 作品背景

### 1、设计开发该作品的目的及意义

随着现代社会人们的生活节奏越来越快，基本上没有时间来看完整的电视剧或者电影，只有通勤路上或者排队路上的短时间，利用这一点时间，用户可以刷短视频来解决短暂的无聊。或者在通勤路上听一听音乐，缓解通勤路上的无聊，本作品就是 mini 音乐视频播放器，可以通过登录验证实现短视频播放和音乐的播放。

### 2、作品现状

市面上有这类作品，它们算法优秀，功能齐全，包含了很多的作品，并且也吸引了很多自媒体博主的加入，规模不断扩大。但是由于功能太多，导致了一些中老年人难以弄懂，无法使用一些功能，通过我们这个 mini 音乐视频播放器通过简单的设计就可以实现出短视频的播放，音乐的播放，用户的登

陆操作等等，简单通俗易懂，开发难度也小。

### 三、 作品设计内容和方法

#### 1、主要设计内容

- 1) 登录 **mini** 音乐视频播放器模块，通过输入已经注册的用户名密码进行登录操作
- 2) 注册用户密码模块，输入未注册的用户名密码进行注册
- 3) 修改密码模块，输入用户名以及原密码新密码进行修改密码操作
- 4) 视频播放器模块，点击播放按钮开始播放，下一个进入下一个视频
- 5) 音乐播放器模块，点击播放开始播放，可循环显示音乐名，下一曲进入下一个音乐
- 6) 我的音乐视频模块，在音乐视频播放器模块点击即可进入查看我的视频和我的音乐

#### 2、解决的关键问题

- 1) 注册账号的问题，要想实现登录验证，每个人有自己的账号，第一步操作就是要进行注册。在注册账号时遇到的问题就是如何避免重复注册账号，在 **localStorage** 中，可以重复的添加密钥，添加相同的密钥，会修改值，这就导致了可能出现注册账号时出现了修改密码的局面。一开始我使用了 **getiteam**，是获取的密码来与输入的账号进行比对是否注册过，导致了一次次的失败，我还百思不得其解，不知道哪里错了。我便想到了通过验证是否存在相同的密钥来实现，但 **localStorage** 没有

提供寻找密钥的方法，最后只能通过获取长度使用 `for` 循环来寻找，最后成功的实现了。



## 账号已注册

重新注册

- 2) 我的音乐视频模块中，布局的问题。由于我的布局知识比较匮乏，对布局不是很懂，导致此部分耗费了较多的时间，在进行查看我的视频我的音乐时，由于是表格，但是没有边框，边框是我自己寻找的背景图片来实现的，所以难以实现居中的效果。在出现很多个字的歌曲名时，会出现对不齐的情况，文字会集体靠右，看着很不美观，最后在网上寻找到相对定位才解决了这个问题。

我的音乐	我的视频
大耳朵图图	广阳岛.MP4
小火车	北滨路.MP4
疯狂果宝	贵阳.MP4
Wanan (Demo)	海.MP4
芳华慢+霜雪千年	晚霞1.MP4
听妈妈讲那过去的事情	晚霞2.MP4

- 3) 在进行登陆，注册，修改密码等操作的时候，会出现没有输入账号密码点击登录如果存在空白账号就可以进行登录的情况，这导致了系统的安全性特别低。在进行思考后，我选择了先获取本地存储的长度，如果等于 0 就会提示用户未注册，表示本地存储里面没有任何数据。在大于 0 的时候，就会判定是否输入了账号密码或者新密码，并且判定输入的用户是否存在。

**此页面显示**

用户不存在，请前往注册

确定

## 修改密码

用户名: 1312

原密码: ...

新密码: ...

确认修改

返回

未大作业/源文件/html/修改密码.html

重庆邮 哔哩叫

**此页面显示**

请输入用户名

确定

## 修改密码

用户名: 请输入用户名

原密码: 请输入原密码

新密码: 请输入新密码

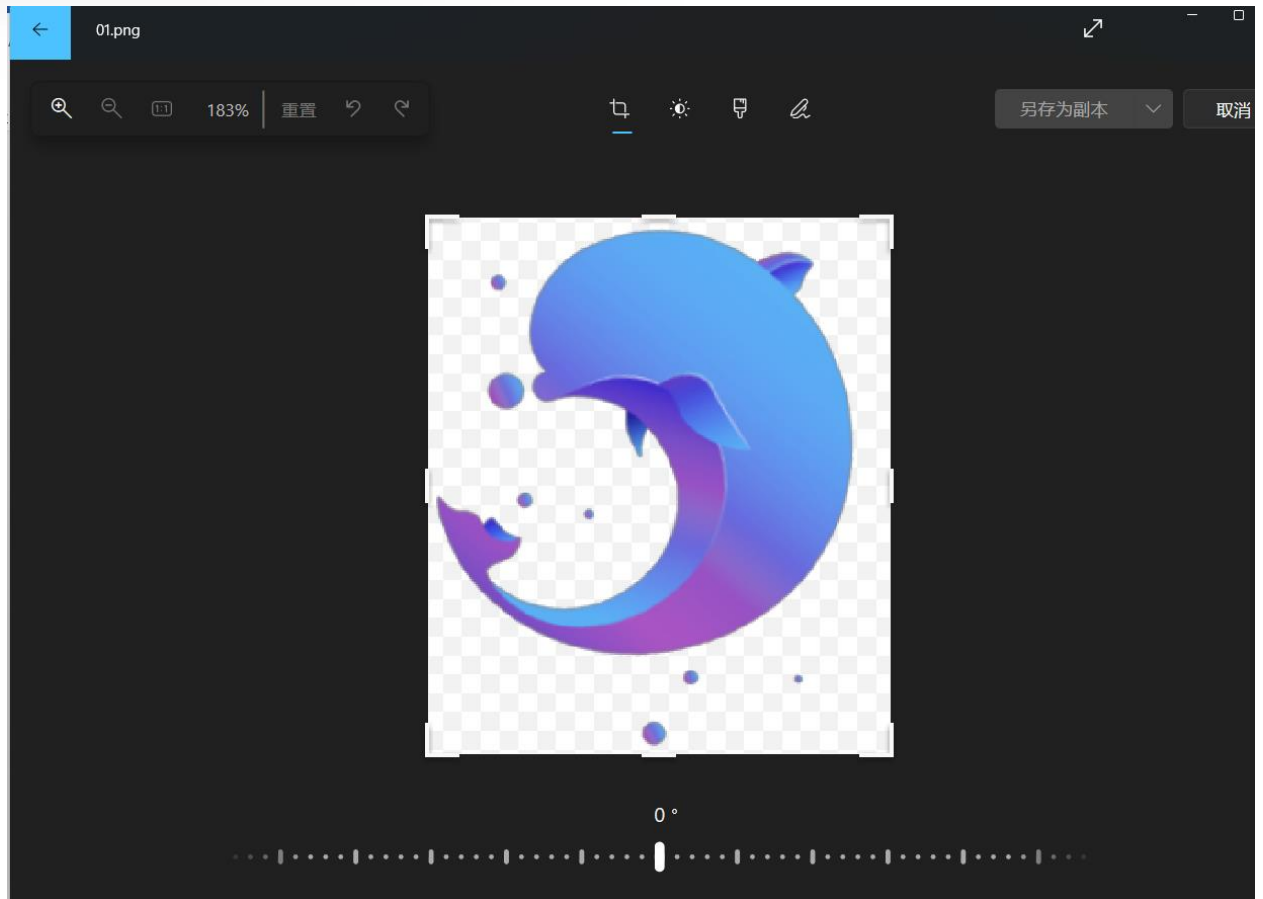
确认修改 返回

- 4) 页面的跳转功能问题，在课程中，只学习了简单的通过超链接加本地的地址来实现跳转，并没有学习通过 `js` 来实现跳转。而这种多个页面的项目，跳转显得十分的重要，没有跳转就无法实现功能，在网上寻找资料后，学习到了很多的跳转方法，最后成功使用了 `self.location =` 来实现跳转，只要两个 `html` 文件在一个文件夹里面就可以实现直接跳转。返回按钮的跳转使用了 `window.history.go(-1)`来实现跳转。



### 3、主要设计方法及采用的辅助工具

在进行一些图片的裁剪时，使用了微软自带的图片剪切裁剪软件进行裁剪，进行了一些按钮图片的裁剪。



在查看注册的账号，修改的密码时，我使用了 **Web** 浏览器调试工具，在里面的应用程序查看了账号密码的存储，修改密码是否成功。还使用了 **web** 浏览器调试工具进行了代码的调试，断点，查看每一行代码的执行步骤，来进行修改优化代码。

### 四、系统功能模块描述及概要设计

1) 登录 **mini** 音乐视频播放器模块，通过输入已经注册的用户名密码进行

## 登录操作

本模块首先判断用户是否输入了用户名和密码，没有输入则会弹出提示框，再判断本地存储是否有数据，输入的用户名是否已经注册，没有数据或者没有注册都会提示“用户不存在”。然后通过用户在登陆界面的 input 输入的已经注册的用户名和密码，再与本地存储里面的密钥和值进行比对判断，正确就跳转到视频播放界面，失败则跳转到登陆失败界面，等待 5 秒会返回登录界面。

### 2) 注册用户密码模块，输入未注册的用户名密码进行注册

本模块首先判断用户是否输入了用户名和密码，没有输入则会弹出提示框，再判断本地存储是否有数据，获取本地存储的长度。然后根据用户输入的用户名判断本地存储是否有相同的用户名，如果存在则跳转到注册失败界面，不存在或者本地存储没有数据则成功注册账号，写入到本地存储里面。

### 3) 修改密码模块，输入用户名以及原密码新密码进行修改密码操作

本模块需要用户在登陆成功后，进入到视频或者音乐播放界面后，才可以点击进入。用户点击进入修改密码模块，输入登陆的账号和旧密码，再输入新密码，系统检测是否输入了账号密码，没有输入则会提示。检测用户输入的用户名是否存在，如果本地存储没有该用户名或者本地存储没有任何数据则会提示。在本地存储进行比对找到账号，直接进行修改，修改成功则跳转到修改成功界面，点击跳转会登录界面重新登录。账号或者密码错误则会提示重新输入。

### 4) 我的音乐视频模块，在音乐视频播放器模块点击即可进入查看我的视

## 频和我的音乐

本模块通过在音乐或者视频播放器点击进入。界面通过三个 `div` 来实现，先是一个大的 `div` 包含两个左右的 `div` 模型，左边包含一些按键和欢迎的字样，右边包含了 `table` 和一些图画，`table` 采用相对定位，来改变之前与背景图片的边框对不齐的情况，垂直方向采用 `vertical-align: middle`; 将其置于中间。数据通过事先写好的数组来导入，每次导入前先判定数组的长度，然后 `innerHTML`，最后 `appendchild`。点击返回页面即可返回到音乐播放器界面或者是视频播放器界面，点击左侧的退出登录可以回到登录界面，点击修改密码即可进入修改密码界面。

### 5) 视频播放器模块，点击播放按钮开始播放，下一个进入下一个视频

本模块通过使用 `h5` 自带的媒体 `api` 来进行设计，直接使用 `controls` 即可给视频添加暂停开始以及音量调节模块，全屏模块。下一个视频时，首先判断当前是否为最后一个视频，最后一个视频则回到第一个视频，否则直接下一个视频，上一个视频时也先判断是否为第一个视频，如果是则回到最后一个视频，否则跳到上一个视频。点击中间的按钮可直接跳转到音乐播放器界面，点击我的音乐视频可以进入我的视频音乐界面。

### 6) 音乐播放器模块，点击播放开始播放，可循环显示音乐名，下一曲进入下一个音乐

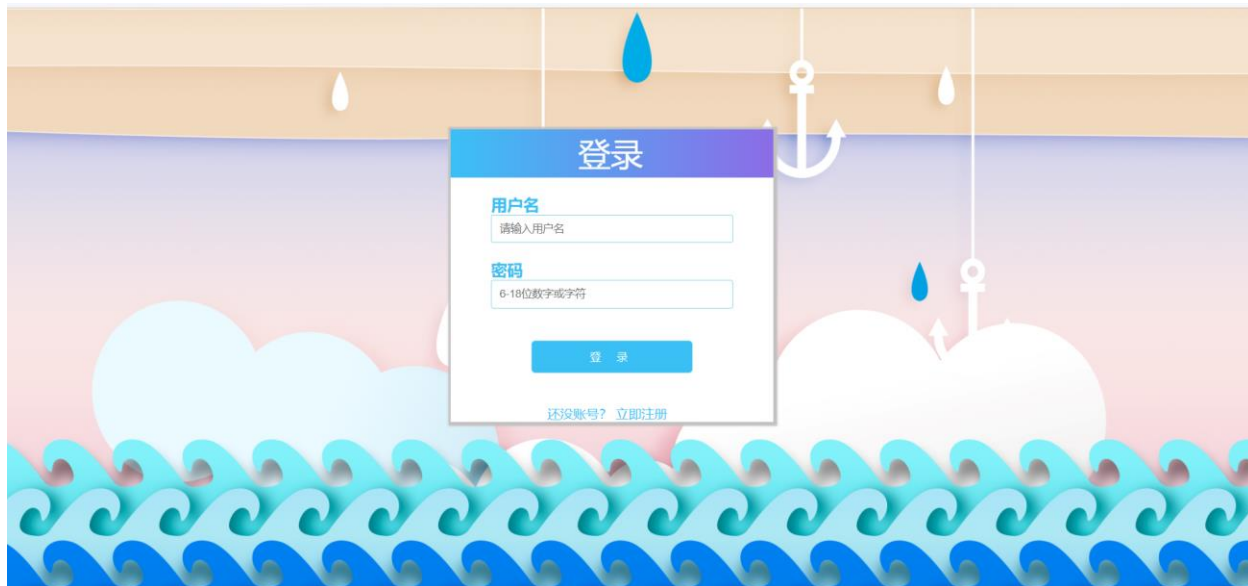
本模块同样也使用了 `h5` 自带的媒体 `api` 来设计实现，调用 `controls` 可以给音乐添加进度调节，暂停播放，音量调节等。下一首音乐时，首先判断当前是否为最后一首音乐，最后一首音乐则回到第一首音乐，

否则直接下一首音乐，上一首音乐时也先判断是否为第一首音乐，如果是则回到最后一首音乐，否则跳到上一首音乐。通过画布 **api** 进行绘画滚动字符，每 **200ms** 执行一次函数，每次将坐标向右移动，移动到一定次数后再往左移动到最初的位置，依次循环。

## 五、系统详细实现与测试结果分析

### 1) 登录 mini 音乐视频播放器模块

使用两个 **input** 来输入登录的用户名和密码，**type** 分别为 **text** 和 **password**，**placeholder** 设为请输入用户名和密码，同时设置一个登陆的按钮，**type** 为 **button**，通过连接设置立即注册跳转的连接，整体页面的框架就设计好了。如下



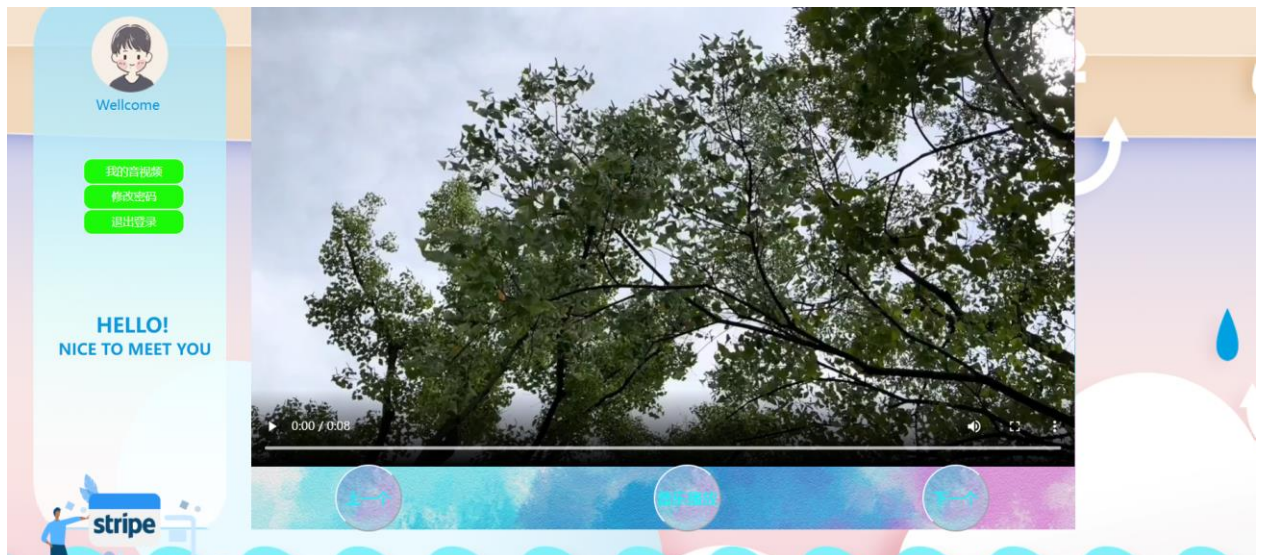
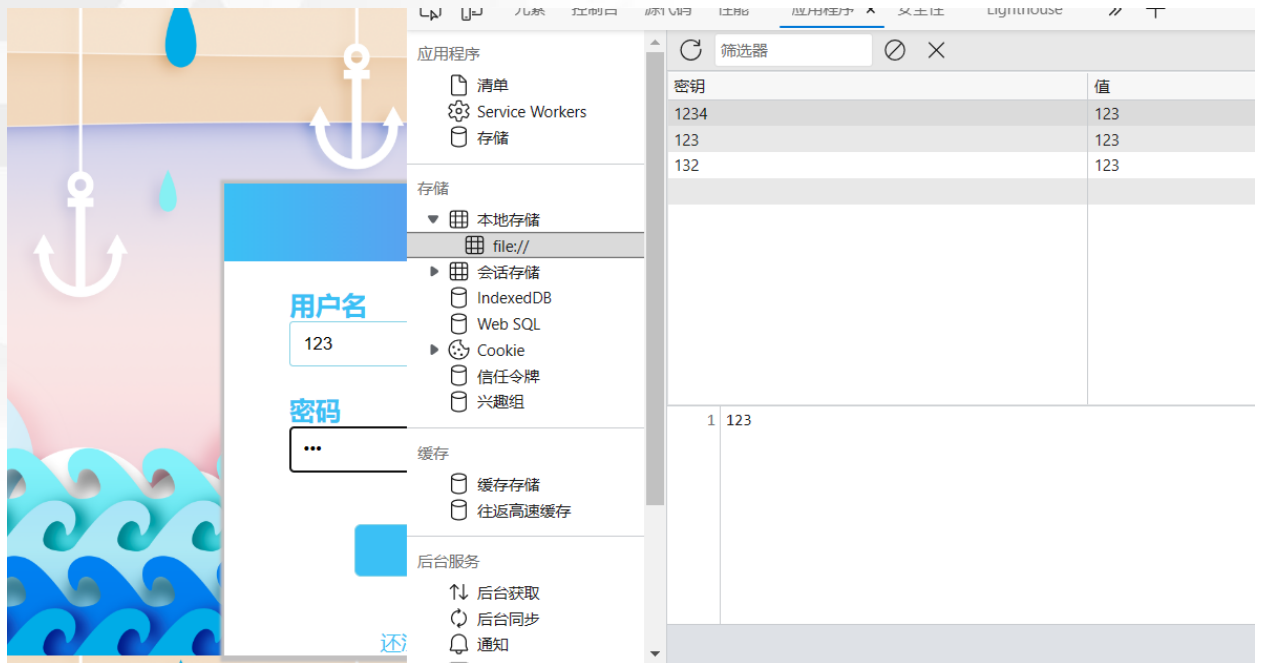
通过获取 **id** 的方式获取用户输入的用户名和密码的 **value**，使用 **if** 检查输入是否为 **0** 表示没有输入内容，则提示请用户输入



如果本地存储的长度为 0 或者不存在则提示用户不存在。



For 循环长度设为本地存储的长度，依次查找 key，找到对应的用户名后开始比对密码，如果正确则跳转到视频播放界面，



错误则跳转到登陆失败界面。



## 2) 注册用户名密码模块

使用两个 `input` 来输入注册的用户名和密码，`type` 分别为 `text` 和 `password`，`placeholder` 设为请输入用户名和密码，同时设置一个注册和返回主界面的按钮，`type` 为 `button`。

用户在登陆前先进行注册操作，用户输入想要注册的用户名以及密码，js 通过 `id` 获取到用户输入的用户名密码的 `value`，判断值是否为 `0`，来判断用户是否输入了用户名密码。

```
var usernamein = document.getElementById("regiusername").value;
var passwordin = document.getElementById("regipassword").value;
if (usernamein == 0) {
    alert("请输入用户名");
    return 0;
}
if (passwordin == 0) {
    alert("请输入密码");
    return 0;
}
```



The image shows a web interface for registering an account. At the top, a white modal dialog box is displayed with the title '此页面显示' (This page displays) and the text '请输入用户名' (Please enter username). It has a blue '确定' (Confirm) button. Below the dialog, the main background is light blue. In the center, the text '注册账号' (Register account) is written in large red characters. Underneath, there are two input fields: '用户名: 请输入用户名' (Username: Please enter username) and '密码: 请输入6~18位数字或字符' (Password: Please enter 6~18 digits or characters). At the bottom, there are two orange buttons: '确认' (Confirm) and '返回' (Return).

此页面显示  
请输入用户名

确定

## 注册账号

用户名: 请输入用户名

密码: 请输入6~18位数字或字符

确认 返回

输入用户名密码后，判断本地存储的长度，是否为 0，如果为 0 即可直接注册，不为 0 就要与本地存储的用户名进行比较，如果已经注册则会提示



账号已注册

重新注册

没有注册则会提示注册成功，点击即可进入登陆界面。



注册成功

返回登录界面

### 3) 修改密码模块

使用三个 `input` 来输入需要修改的用户名和旧密码新密码，`type` 分别为 `text` 和 `password`，`password`，`placeholder` 设为请输入用户名和密码，同时设置一个确认修改和返回音乐视频界面的按钮，`type` 为 `button`。背景图采用与注册界面相同的来进行设计。

用户在输入正确的账号密码登陆成功进入音乐视频播放界面后，点击左侧的修改密码按钮即可进入，用户输入想要修改的用户名以及密码，js 通过 `id` 获取到用户输入的用户名密码的 `value`，判断值是否为 0，来判断用户是否输入了用户名密码。

```
function update() {
    var usernameup = document.getElementById("usernameup").value;
    var passwordup = document.getElementById("passwordup").value;
    var passworduse = document.getElementById("passworduse").value;
    var leng = window.localStorage.length;
    if (usernameup == 0) {
        alert("请输入用户名");
        return 0;
    }
    if (passworduse == 0) {
        alert("请输入原密码");
        return 0;
    }
    if (passwordup == 0) {
        alert("请输入新密码");
        return 0;
    }
}
```

入作业/源文件/html/修改密码.html

此页面显示

请输入原密码

确定

## 修改密码

用户名: 123

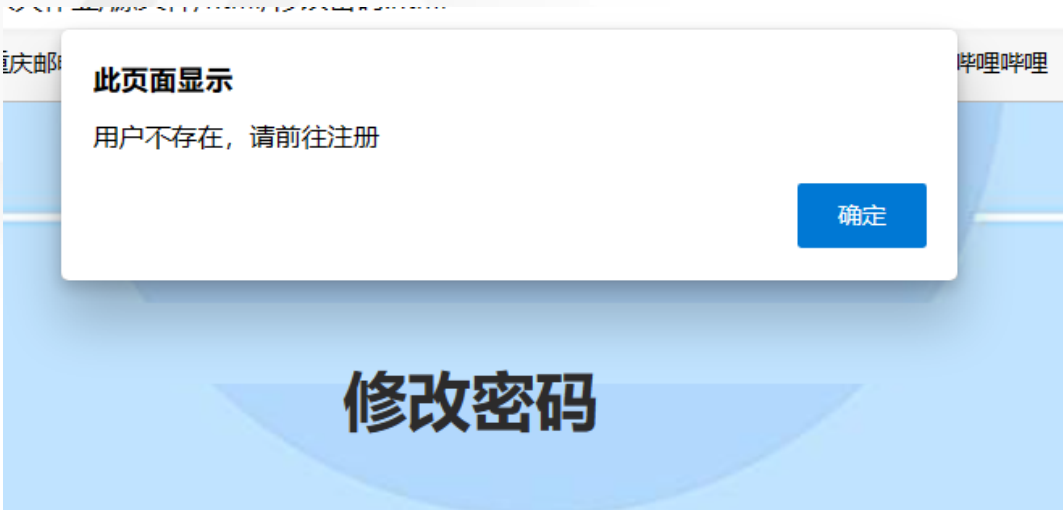
原密码: 请输入原密码

新密码: 请输入新密码

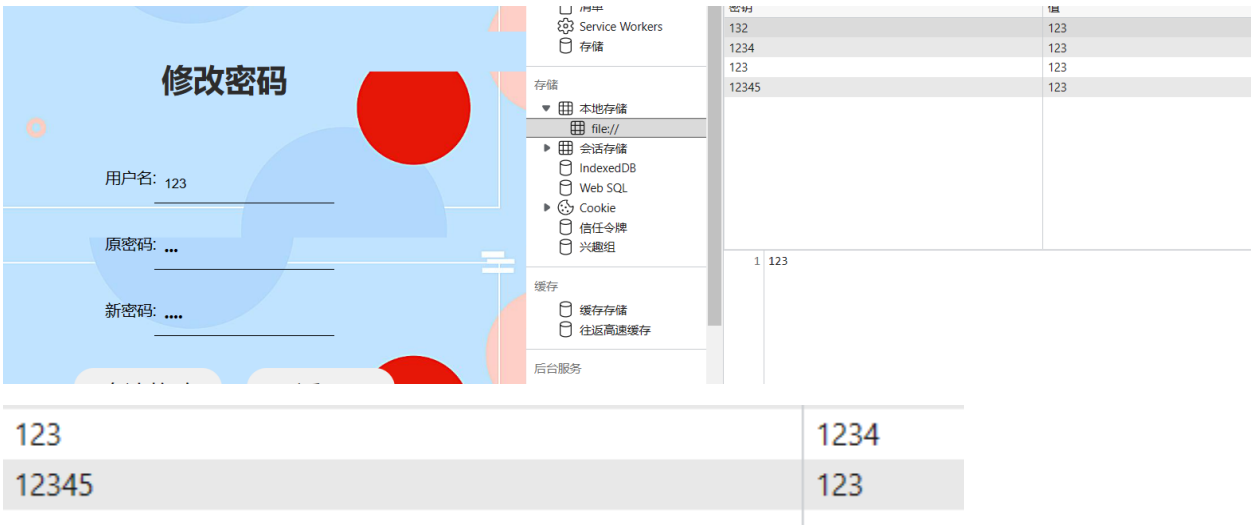
确认修改

返回

先获取本地存储的长度，检测长度是否为 0，如果为 0 则表示没有存储任何数据，则提示用户不存在。



以 `leng` 为循环次数进入循环，查找与用户输入用户名相同的用户名，检查原密码是否相同，相同则修改为新密码。





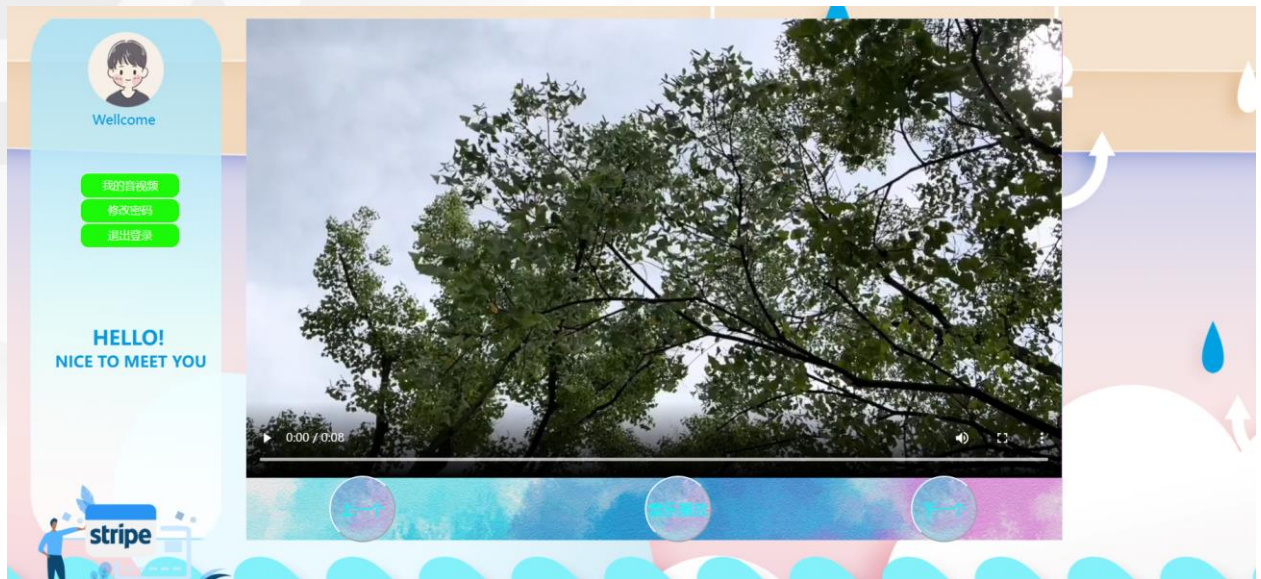
## 修改成功!

返回登录

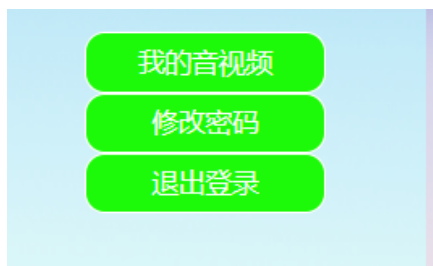
点击返回按钮可以通过 `history.go(-1)` 回到刚刚的页面。

#### 4) 视频播放器模块

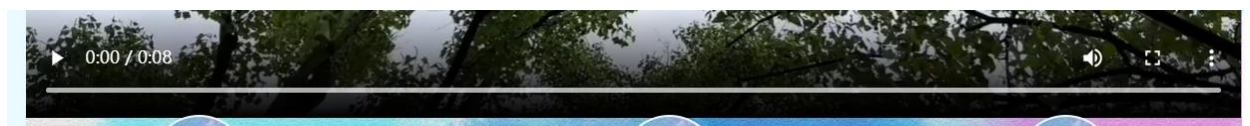
本模块通过登陆界面登录成功后进入。最后三个界面均是通过三个 `div` 来实现，先是一个大的 `div` 包含两个左右的 `div` 模型，左边包含一些按键和欢迎的字样，右边包含了视频和一些图画，以及位于最下方的三个按钮，三个按钮都通过 `margin` 来实现定位。



点击左侧的按钮，通过 `self.location` 即可进入同目录下面的 `html` 界面，分别为修改密码，查看我的音乐视频，退出到登录界面。



视频通过 `src` 导入，在标签中加入 `controls` 来实现播放暂停，音量以及全屏。



音乐播放按钮也是通过 `self.location` 来跳转到音乐播放界面的。

点击下一个视频，首先通过 `id` 获取到 `video`，如何判断当前播放的为第几个视频，如果为最后一个则跳转回第一个，否则直接下一个，然后再重新播放。





上一个也是同样的道理，如果为第一个则回到最后一个。

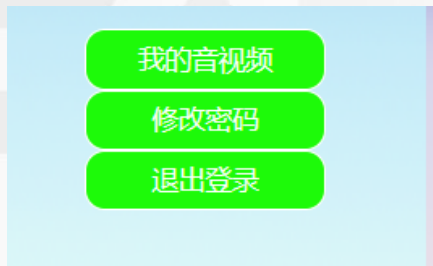
## 5) 音乐播放器模块

本模块通过视频播放界面点击后进入。最后三个界面均是通过三个 **div** 来实现，先是一个大的 **div** 包含两个左右的 **div** 模型，左边包含一些按键和欢迎的字样，右边包含了显示歌曲名的画布和音乐控制器，以及位于最下方的三个按钮，三个按钮都通过 **margin** 来实现定位。

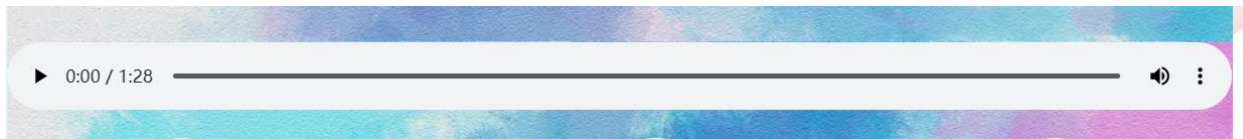


左侧同样也有三个相同的按钮，通过 **self.location** 即可进入同目录下面的 **html** 界面，分别为修改密码，查看我的音乐视频，退出到登录界面。





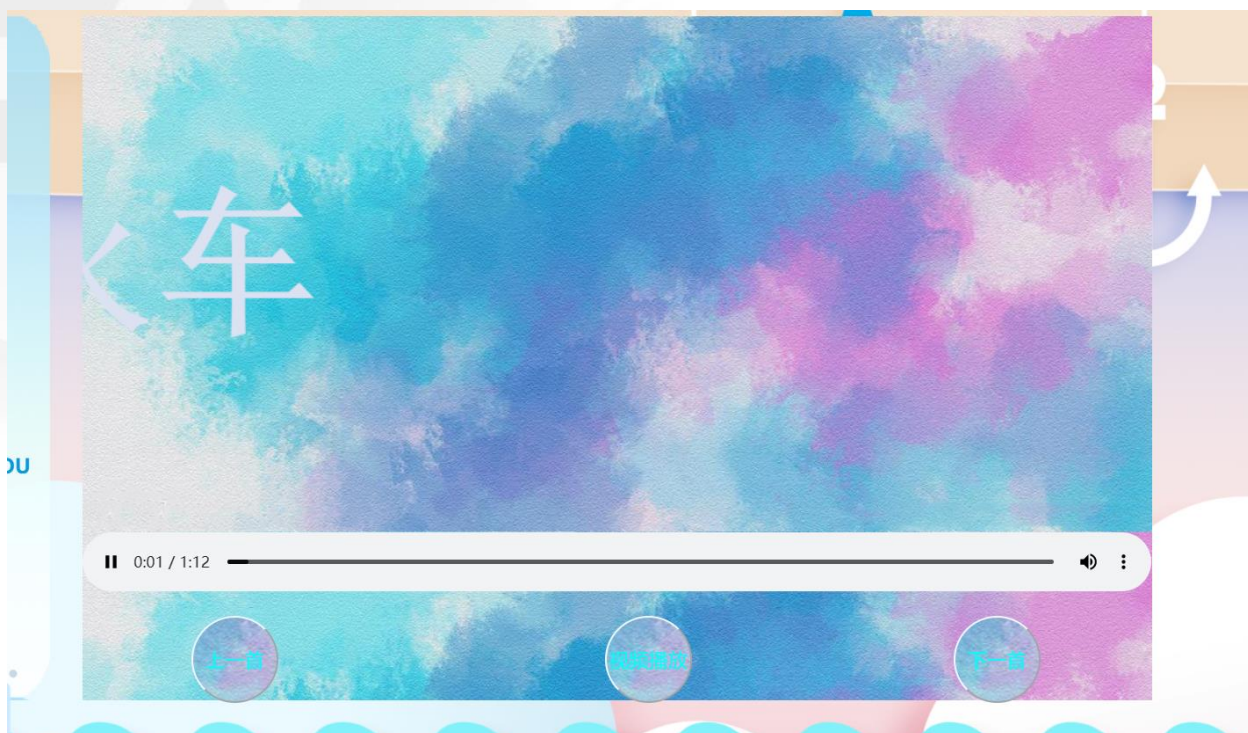
音乐通过 `src` 导入，在标签中加入 `controls` 来实现播放暂停，音量以及全屏。



视频播放按钮也是通过 `self.location` 来跳转到音乐播放界面的。

点击下一首歌曲，首先通过 `id` 获取到 `audio`，如何判断当前播放的为第几个视频，如果为最后一首歌曲则跳转回第一首歌曲，否则直接下一首，然后再重新播放。





## 6) 我的音乐视频模块

本模块通过音乐或者视频播放界面点击后进入。最后三个界面均是通过三个 **div** 来实现，先是一个大的 **div** 包含两个左右的 **div** 模型，左边包含一些按键和欢迎的字样，右边包含了 **table** 和一些图画，**table** 采用相对定位，来改变之前与背景图片的边框对不齐的情况，垂直方向采用 **vertical-align: middle;** 将其置于中间。

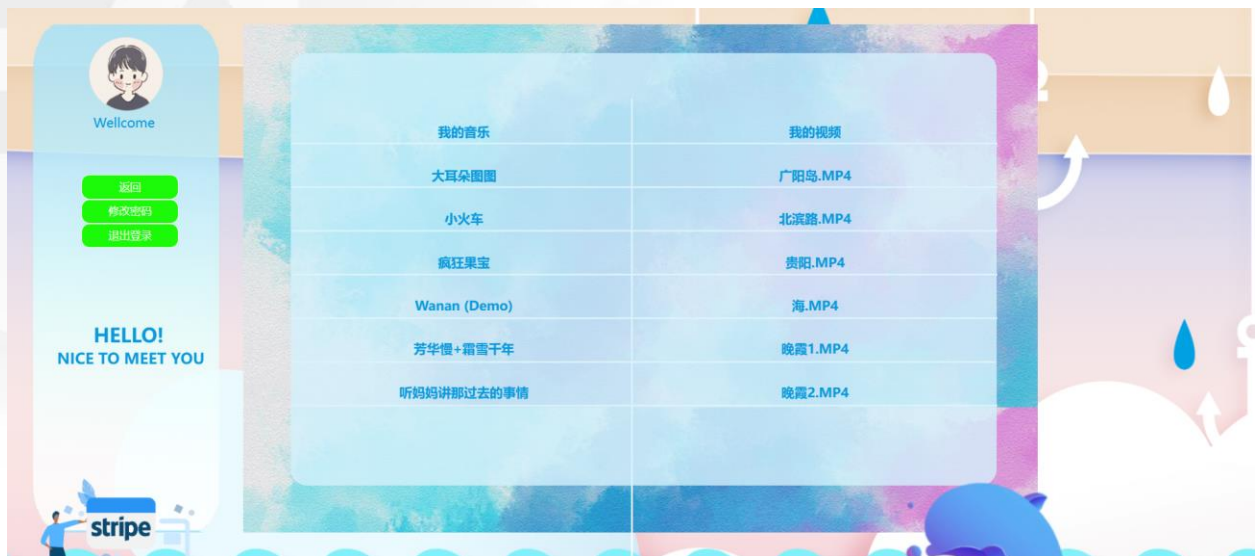


Table 的头选择了在 `html` 中直接写出，而数据选择通过 `js` 来进行动态的修改，根据用户的音乐和视频来进行动态的调整。先通过 `id` 获取到 `table`，通过 `for` 循环 `innerHTML` 依次插入用户的音乐和视频名称，每次都 `appendChild (td)`，最后再 `appendChild (tr)` 直到循环结束插入。

## 六、问题与总结

本作品存在如下的问题：存储用户名密码的本地存储太过简单简陋，可以被用户所看到安全性较低。由于此次项目只限于了 `html`，`css`，`JavaScript` 的简单前端知识，学习得还不够完善，所以在以后的学习中，该作品可以使用后端 `java`，`php`，`go`，`c` 语言来实现与数据库的连接，通过后端来将用户名以及密码存储到后端数据库，安全性更高。每个用户的视频没有联网存储，只是在本地存储的，每个用户都是一样的，该问题也是需要后端来与数据库建立连接，存储在服务器里面，来达到每个用户可以自行添加视频音乐，下次登陆的时候还可以进行播放已经添加的音乐视频。